# Team Project: Deliverable 1 - Proposal CSCE 5430 (Spring 2022)

**Deliverable I: Project Proposal including Project Plan and Risk Management**

1. **A project title, group name, and group members’ names.**

**Project Title:** The Bug Tracking System

**Group Name:** BUG MAFIA

**Group Members:**

1. Neha Goud Baddam
2. Vamsi Sai Konidena
3. Reshmi Chowdary Divi
4. Prem Kumar Maddula
5. Rahul Mandaloju
6. Harika Uppalapati
7. Purandhara Maharshi Chidurala
8. Shiva Surya Vardhan Reddy
9. **A project description: a detailed description of the system you plan to develop   
   including development environments (language, platform, and so on)**

**Project Description:**

A bug tracking system is a software application that maintains track of reported software issues in software development projects. Most bug tracking systems allow for the triage of incoming problems, which entails assigning a bug's priority (and maybe severity) and assigning it to a specific developer. Many bug tracking systems, including those used by most open-source software projects, allow end-users to enter bug reports directly. A bug tracking system is a database that allows you to keep track of defects (and frequently suggestions) in your software project.

**Languages:**

Although there are hundreds of different programming languages available today, most of the software we use is written in only eight of them. Each language is distinct and has its own characteristics. The languages used by the bug tracking system are as follows; python, SQL, HTML, Oracle DB, CSS, and JavaScript.

Firebase - For Authentication and Database

1. **An initial planning that lays out the big timeline for your project and expected milestones. Include Gantt and/or PERT charts where appropriate. This will be modified in later deliverables. At this point, you do not have enough details to develop a complete plan, but you can do a high-level planning exercise. Please refer to the deliverable deadlines posted on the Canvas to make a plan. Use KanBan board to track the project schedule including each member’s status.**

Deliverable 1 : Project Proposal and Risk Analysis

Deliverable 2 : Requirement Gathering and System Specification

Deliverable 3 : Prototype Version -1 ( System Design, Coding and Initial Testing)

Deliverable 4 : Prototype Version – 2 (Fixing existing bugs)

Deliverable 5 : Final Product - Fixing the Bugs from previous deliverables (Final testing and Documentation)

**Graphical user interface, application

Description automatically generated**

**Timeline

Description automatically generated**

**Diagram

Description automatically generated**

1. Risk management (content described above).

**Risk Management:**

Many of the downsides of defect tracking have less to do with defect tracking and more to do with the overhead of the processes and tools. Some companies utilize various technologies to track multiple faults, and these systems frequently don't work well together. Though defect-tracking software aids in the storage of paperwork, it can also obstruct communication by preventing team members from conversing and contributing. There are sure to be misconceptions when code is tossed "over the wall" from development to test, and the flaws are thrown back to development, and the sole communication regarding defects is done through a tool.

However, if a tool has been chosen, it would be beneficial for someone on the team to become a super-user and modify the device to make it as user-friendly as possible for the team, possibly by adding templates or prefabricated reports.

**Contingency plans for risks:**

Teams must collaborate to determine when defects will be monitored and any related processes or criteria, such as the amount of information that must be tracked for each ticket. Improve the communication, flow, and quality of these procedures. Continuously analyze your procedures during retrospectives and change them over time.

Every software development team should have a process to deal with faults. Whether the tools are as simple as sticky notes or as complicated as an entire application, lifecycle management tools should be figured out how to use them and build practices that will work best for you as a team.

1. **A section that describes each team member’s roles for the project.**
2. Neha Goud Baddam: Full Stack
3. Reshmi Chowdary Divi: Back-end Developer
4. Prem Kumar Maddula: Back-end Developer
5. Rahul Mandaloju: Frontend Developer
6. Vamsi Sai Konidena: Frontend Developer
7. Harika Uppalapati: UI Developer
8. Purandhara Maharshi Chidurala: UI Developer
9. Shiva Surya Vardhan Reddy: Testing and Documentation
10. **Member contribution table (should describe who wrote what parts of the report).   
    Add more rows as needed.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Member Name** | **Contribution Description** | **Overall Contribution (%)** | **Note (If Applicable)** |
| **Neha Goud Baddam** | Document – Deliverable 1 and PPT | 15 |  |
| **Vamsi Sai Konidena** | Risk Management | 14 |  |
| **Reshmi Chowdary Divi** | Project Description and languages | 15 |  |
| **Prem Kumar Maddula** | Risk Management | 14 |  |
| **Rahul Mandaloju** | Member’s role and Contribution Table | 14 |  |
| **Harika Uppalapati** | Timeline for the project | 14 |  |
| **Purandhara Maharshi Chidurala** | Gantt Charts and PERT Charts | 14 |  |
| **Shiva Surya Vardhan Reddy** | PPT | 14 |  |